

Remarks

Status of application

Claims 1-45 were examined and stand rejected in view of prior art. The distinctions between Applicant's invention and the cited prior art references are discussed in detail in the following remarks. In view of the below remarks, reexamination and reconsideration are respectfully requested.

The invention

Applicant's invention comprises a database system and methodology providing extended memory support. Applicant's methodology for extended memory support in a database system provides for creating a secondary cache in memory available to the database system using shared memory file system and memory mapped file features of an operating system. A virtual address range is mapped to at least a portion of the secondary cache. When the primary cache is full pages are replaced from the primary cache to the secondary cache. When a particular page is requested, the secondary cache is searched if the page is not found in the primary cache and if the particular page is found in the secondary cache its virtual address in the secondary cache is determined based on the mapping. The particular page found in the secondary cache is then swapped with a page in the primary cache, so as to replace a page in the primary cache with the particular page from the secondary cache.

Prior art rejections

A. Section 103(a) rejection: Alsup in view of Darcy

Claims 1-3, 5-26, and 28-45 stand rejected under 35 U.S.C. 103(a) as being anticipated by US Published Application No.: 2004/0103251 A1 to Alsup (hereinafter "Alsup") in view of US Patent No: 7,124,249 BI to Darcy (hereinafter "Darcy"). Here, the Examiner continues to rely on Alsup's L2 microprocessor cache as substantially teaching Applicant's claimed invention, but acknowledges that Alsup does not teach the following elements of Applicant's claims: using a memory mapped file, creating a secondary cache in system memory available to the database system; mapping a virtual address range to at least a portion of the secondary cache if the particular database page is

found in the secondary cache, and determining a virtual address in the secondary cache where the particular database page resides based on the mapping. The Examiner adds Darcy for these teachings.

At the outset, Applicant's claims, as amended by the amendment previously filed by Applicant in response to the first office action, make it clear that Applicant's claimed invention is a software-implemented cache mechanism that provides for creation of a cache in ordinary system memory (i.e., memory located external to the microprocessor). As such, it is clearly distinguishable from Alsup's L2 cache physically burned onto a microprocessor chip, which is a hardware solution supporting operations of the microprocessor. More particularly, Applicant's claimed invention specifies the secondary cache is created in system memory using a memory mapped file (i.e., an operating system-supported shared memory file system), thereby extending available system memory (e.g., from 4GB to 64GB) without any specific hardware support. This is described, for example, in Applicant's specification as follows:

... the present invention makes use of a shared memory file system feature ("shmfs") and a memory mapped file feature ("mmap") available on Linux platforms. A shmfs file is created and the database system maps a virtual address range (window) to the portion of the shmfs file through the mmap interface. This mechanism enables the database system to address a portion of memory mapped shmfs file whose size can be as large as 64G.

(Applicant's specification, paragraph [0067]).

To the extent that the data used by the database system is not available in the primary cache (especially, due to a cache size constrained to less than 4GB on 32-bit machines), the use of a memory mapped file space to create a secondary cache from which the database system replaces database pages from the primary cache enables the database system to address a portion of memory mapped file whose size can be as large as 64G (e.g., in an embodiment operating on a 32-bit platform). This enables the database system to avoid disk reads while making effective use of memory mapping and shared memory features of the operating system. This feature is also specifically included in Applicant's claims including, for instance, the following limitations of claim 1:

A method for extended memory support in a database system having a primary cache for storing database pages, the method comprising:
using a memory mapped file, creating a secondary cache in system memory available to the database system;
mapping a virtual address range to at least a portion of the secondary cache;

(Applicant's claim 1, emphasis added)

The Examiner acknowledges that Alsup does not teach using a memory mapped file to create a secondary cache in system memory available to the database system and references Darcy (at column 5, lines 9-16) for these teachings. However, Applicant's review of the referenced portion of Darcy (as well as the balance of the reference) finds no mention of creating the secondary cache using a memory mapped file. Rather the portion of Darcy referenced by the Examiner simply provides as follows:

The computer comprises a processor programmed to implement a software cache hierarchy having at least two software caches that are interrelated to form the cache hierarchy, the at least two software caches including at least a first software cache and a second software cache, wherein the first and second software caches employ different hashing techniques for mapping an address into the first and second software caches.

(Darcy, col. 5, lines 9-16).

As illustrated above, Darcy simply describes a software cache hierarchy including two or more software caches. Darcy does not include the specific teachings of Applicant's specification and claims of creating a secondary cache using a memory mapped file feature. Moreover, the "caches" described by Darcy include not only caches created in system memory, but Darcy also provides that caches may be implemented using "less expensive storage resources (e.g., disks)" (Darcy, col. 30, lines 6-11). This is in direct contrast with Applicant's approach which provides extended memory support in system memory in order to avoid having to read in database pages from disk storage.

Neither Alsup nor Darcy addresses the above identified problem that is solved by Applicant's invention. Specifically, in certain architectural scenarios, such as Linux running on a 32-bit Intel microprocessor, it is desirable to support a very large address space (e.g., 64GB) even though the platform itself only supports a 4GB memory space. Applicant's invention provides an approach for mapping to extended memory on various

devices so that an application can bring data in from various sources without needing to know or be concerned about the underlying location of this data. Creating a secondary cache in system memory using a memory mapped file is particularly important in for large database systems as it enables the system to address more memory than can be supported in the address space of a given platform (e.g., a 32-bit platform).

Applicant's solution also includes a data structure that maintains a mapping to pages in the secondary cache (Applicant's specification, paragraphs [0089]-[0091]). In addition to providing a translation from page name to an offset, the offset in the data structure is also a mapping which enables the system to find the actual location of where a particular page is stored in the shared cache (Applicant's specification, paragraph [0092]). In this manner, an application can simply request a desired database page (e.g., page P2) and Applicant's system handles the process of searching for the desired page and making it available, regardless of its physical location. The page may, for example, be brought in from memory of a remote (i.e., different) machine in a database cluster. These features are also describe in Applicant's claims, including, for example, in the following limitations of claim 1:

mapping a virtual address range to at least a portion of the secondary cache;
when the primary cache is full, replacing database pages from the primary cache using the secondary cache;
in response to a request for a particular database page, searching for the database particular page in the secondary cache if the particular database page is not found in the primary cache;
if the particular database page is found in the secondary cache, determining a virtual address in the secondary cache where the particular database page resides based on the mapping;

(Applicant's claim 1, emphasis added)

Furthermore, the focus of Applicant's invention is to provide extended memory support for caching database pages (e.g., for tables, indexes and the like) in system memory in order to avoid having to bring this data into memory from disk in order to perform runtime operations on the data. The caches described Alsup and Darcy, in contrast, are very generic and are not suitable for supporting the transactional requirements of database systems.

Further distinctions between Applicant's solution and the prior art references are also shown in Applicant's dependent claims. For example, Applicant's claims 2 and 3 include claim limitations of creating the secondary cache using a shared memory file system provided by an operating system. Here, the Examiner references Alsup and Darcy for these teachings as follows:

As to claim 2 Alsup discloses wherein said creating step includes creating the secondary cache. The Darcy reference teaches wherein said creating step includes creating the secondary cache (see figure 1, L2, which communicates with Li data cache 101 B and another shared memory file.

As to claim 3, Alsup discloses wherein the shared memory file system is available as part of an operating system on a computer platform on which the database system is running (see figure 2, system memory communicates with the data base, as part of operating system.

As illustrated above, the Examiner references Fig. 1 and Fig. 2 of Alsup as including the teachings of creating the secondary cache using a shared memory file system. However, Fig. 1 of Alsup clearly shows that the L2 secondary cache (as well as the L1 primary cache) is located on the Microprocessor 100 (i.e., implemented in hardware). In addition, neither Fig. 1 nor Fig. 2 of Alsup includes any reference whatsoever to a shared memory file system (whether provided as part of an operating system or otherwise). As Alsup's L2 cache is implemented in hardware (i.e., on a microchip) it obviously cannot be created using an operating system shared memory file system. Applicant's detailed review of the Darcy reference also finds no mention of a shared memory file system or of creating a secondary cache using an operating system's shared memory file system.

Another distinction is illustrated by Applicant's claim 18, which includes claim limitations of determining database pages to be maintained in the secondary cache based, at least in part, on workload of the database system. Applicant's invention retains frequently used runtime data objects (e.g., database tables or indexes) in cache in system memory, so that those data objects are readily accessible to support the specific transactional needs required in a database system. Only certain types of database objects are stored in the secondary cache of Applicant's invention as Applicant's invention selectively determines the appropriate items to be brought into the case based on the

workload of the database system (Applicant's specification, paragraph [0093]). The Examiner again references Fig. 2 of Alsup for the corresponding teaching; however, Fig. 2 simply shows the L1 and L2 microprocessor cache arrangement and does not even make mention of a database system or its workload. Thus, the claims are believed to be allowable for these additional reasons.

For the reasons stated above, it is respectfully submitted that neither Alsup nor Darcy, either alone or in combination, teach or suggest all of the limitations Applicant's claims. Accordingly, it is believed that the claims 1-3, 5-26, and 28-45 distinguish over the cited art and that the rejection under Section 103(a) is overcome.

B. Section 103(a) rejection: Darcy in view of Austin

Claims 4 and 27 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Darcy (above) in view of Austin et al., (US No: 20030162544 A1, hereinafter "Austin"). Here, the Examiner relies on Darcy as teaching all of the limitations of claims 1-3 (discussed above), but adds Austin for the proposition that it teaches use of the use of a Linux operating system. The claims are believed to be allowable for at least the reasons cited above pertaining to the Section 103 rejection based on Alsup and Darcy. Nothing in Austin cures the above-described deficiencies of Darcy (and Alsup) as to Applicant's claimed invention.

As described above in the discussion of the Section 103 rejection (as well as in Applicant's response to the first office action), Applicant's claimed secondary cache is created using an operating system's shared memory file system (as well as its memory mapped file feature). These teachings of creating a software-implemented cache using specific operating system features are very different than Alsup's on-board processor cache. In addition, although Darcy discusses software-implemented caches, Darcy makes no mention of creating a cache using these shared memory file system and memory mapped file features of an operating system as provided in Applicant's specification and claims.

Claim 4 (when read in conjunction with claims 1-3) does not simply describe use of the Linux operating system (as taught by Austin), but instead specifies a software-implemented secondary cache created in system memory using the Linux shared memory

file system (as well as using Linux memory mapped file feature). Claim 27 includes similar limitations. These claim limitations of a software-based secondary cache which is created using the Linux operating system's shared memory file system and memory mapped file features are not taught or suggested by the combination of Alsup, Darcy and Austin. Accordingly, claims 4 and 27 are believed to distinguish over the combined references, and that any rejection under Section 103 is overcome.

Any dependent claims not explicitly discussed are believed to be allowable by virtue of dependency from Applicant's independent claims, as discussed in detail above.

Conclusion

In view of the foregoing remarks, it is believed that all claims are now in condition for allowance. Hence, it is respectfully requested that the application be passed to issue at an early date.

If for any reason the Examiner feels that a telephone conference would in any way expedite prosecution of the subject application, the Examiner is invited to telephone the undersigned at 925 465 0361.

Respectfully submitted,

Date: August 3, 2007

/G. Mack Riddle

G. Mack Riddle, Reg. No. 55,572
Attorney of Record

925 465-0361
925 465-8143 FAX